

### **REMARKS**

Claims 1-20 were pending at the time of examination. Claims 1-4, 8, 10, 12, and 14-20 have been amended. No new matter has been added. The applicant respectfully requests reconsideration based on the foregoing amendments and these remarks.

### **Objections to the Claims**

Claim 12 was objected to because of a misspelling of the term “instance” in two places. In order to overcome the objections, the applicant has corrected the spelling and submits that claim 12 is thus in allowable form.

### **Claim Rejections – 35 U.S.C. § 103**

Claims 1-20 were rejected under 35 U.S.C § 103(a) as being unpatentable over U.S. Patent No. 6,041,180 to Perks et al. (hereinafter “Perks”) in view of U.S. Patent No. 6,308,320 to Burch (hereinafter “Burch”). The applicant respectfully traverses these rejections.

Claim 1, as amended, recites the step “identifying one or more instances available for use in the one or more library object files, using linker symbol names for the one or more instances;”. The Examiner states that this step is shown in Perks and alleges that “instances STK1.OBJ and STK2.OBJ of class templates are equivalent to identifying instances available the linking process.” The applicant respectfully disagrees. The STK1.OBJ and STK2.OBJ files are object files, which result from compiling a template file STK.CPP (*see* Perks, col. 6, lines 12-35). Thus, the STK1.OBJ and STK2.OBJ object files can correctly be viewed as instances of a class template file. However, the STK1.OBJ and STK2.OBJ object files are created through compilation of a template file, and are not instances identified in one or more library object files, as required by claim 1.

In fact, one of the main advantages of the applicant’s invention is to reduce compilation time by not having to create every instance (such as the STK1.OBJ and STK2.OBJ object files) through compilation, but instead retrieving available instances from the libraries without any significant amount of preparatory work and/or costly operations (*see* specification page 7, line 24 – page 8, line 2). This is made possible through the use of linker symbol names to identify the instances.

Claim 1 further requires the step of “determining whether the first instance has been identified in the one or more library object files.” Since no identification of instances takes place

in Perks, per the above discussion, it is also clear that Perks cannot show “determining whether the first instance has been identified...”. The only determination that is made in Perks is whether there is any common code in the instance object files, which is made during the generation of the executable file in the linking phase (see Perks, col. 6, lines 65-67). This comparison of program code in object files is clearly different from “determining whether the first instance has been identified in the one or more library object files,” as required by claim 1.

The Examiner acknowledges that Perks does not show libraries, and relies on Burch for this showing. Burch discloses a “reuse depository,” which is a directory that contains compiled object files that will be reused (see Burch, col. 10, lines 32-34). That is, the reuse depository is a collection of object files. This is different from the applicant’s invention, which recites one or more library object files that each contains instances available for use, as required by claim 1. Furthermore, in the applicant’s invention, the library object files are directly accessible by the compiler (*see* specification, page 7, line 27 – page 8, line 1), which is made possible through the use of linker symbols. The reuse depository in Burch, on the other hand, does not include any linker symbols and therefore has to be accessed using an “intermediate code generator” (*see* Burch, col.10, lines 34-39). Consequently, also no time savings of the type that is accomplished in the applicant’s invention is possible in Burch. For at least these reasons, there would be no motivation or desire to combine Perks and Burch. Furthermore, there can be no reasonable expectation of success, which is required to be shown by the Examiner in order to establish a *prima facie* case of obviousness, since the reuse depository in Burch requires that there be an intermediate code generator available. No such intermediate code generator exists in Perks. For at least these reasons, the rejection of claim 1 is unsupported by the art and should be withdrawn.

Claims 2-9 are all dependent directly or indirectly from claim 1, and the rejection of these claims is unsupported by the cited art for at least the reasons discussed above with regards to claim 1, and should be withdrawn.

Claim 10, as amended includes “a library object file including at least one instance available for use by the source program, the at least one instance being identifiable by a linker symbol name;” and “an enhanced compiler suitable for compilation of source code, wherein the enhanced compiler accesses the library object file to identify the one instance available in the library object file.” Both of these limitations recite library object files and linker symbol names, which are neither shown in Perks nor in Burch, as discussed above with regards to claim 1. As the same rationale was used in rejecting claim 10 as in rejecting claim 1, the discussion above

with regards to claim 1 applies also to claim 10, and the rejection should be removed for at least the same reasons.

Claims 11-13 are all dependent directly or indirectly from claim 10, and the rejection of these claims is unsupported by the cited art for at least the reasons discussed above with regards to claim 10, and should be withdrawn.

Claim 14, as amended, recites the step: “examining a linker name table of the one or more associated library object files;” which the Examiner considers to be equivalent to examining the CRC codes and associated comment records in the object files in Perks. The applicant respectfully disagrees. The linker names in the linker name table include linker names of instances that are available in the library (*see* specification page 10, lines 31-32). In Perks, the CRC codes and associated comment records correspond to individual object files that have been generated in a conventional compilation process, and do not represent names of instances that are available in a library. The same reasoning applies to the “extracting...” and “storing...” steps of claim 14.

Claim 14 further recites “comparing the first linker symbol name with the one or more stored linker symbol names;”. The Examiner states that “aliasing of names with an associated CRC is equivalent to comparing names of stored symbol names and a requested linker symbol name.” The applicant respectfully disagrees. Again, in Perks, the CRC codes and associated comment records correspond to individual object files that have been generated in a conventional compilation process. A comparison is made between such object files in order to get rid of redundant program code. In the applicant’s invention, a first linker symbol name, received in a request, is compared with one or more stored linker symbol names from a library object file, to determine whether the first symbol name has already been stored.

Finally, claim 14 recites “creating the first instance only when said comparing indicates that the first linker symbol name is not one of the stored linker symbol names.” This is not shown in Perks, where all instances are created through compilation. For at least these reasons, the rejection of claim 14 is unsupported by the art and should be withdrawn.

Claims 15-16 both depend from claim 14, and the rejection of these claims is unsupported by the cited art for at least the reasons discussed above with regards to claim 14, and should be withdrawn.

Claim 17 is a *Beauregard* claim corresponding to claims 1 and 2. For reasons substantially similar to those set forth above with regards to claim 1 and 2, the applicant

respectfully contends that the rejection of claim 17 is unsupported by the cited art and should be withdrawn.

Claims 18-20 all depend from claim 17, and are *Beauregard* claims corresponding to claims 3, 4, and 6, respectively. The rejection of these claims is unsupported by the cited art for at least the reasons discussed above with regards to claim 17, and should be withdrawn.

### Conclusion

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP

A handwritten signature in black ink, appearing to read "Fredrik Mollborn", written in a cursive style.

Fredrik Mollborn  
Reg. No. 48,587

P.O. Box 778  
Berkeley, CA 94704-0778  
(650) 961-8300